일반논문 (Regular Paper)

방송공학회논문지 제17권 제6호, 2012년 11월 (JBE Vol. 17, No. 6, November 2012)

<http://dx.doi.org/10.5909/JBE.2012.17.6.1014>

ISSN 1226-7953(Print)

MPEG-4 LAsER PMSI를 활용한 Secondary Device 기반 파노라믹 영상 슬라이드 쇼 재생 기술

박 용 철^{a)}, 김 규 현^{a)†}

Slide-show of Panoramic Image through a Secondary Device by using MPEG-4 LAsER PMSI

Yongchul Park^{a)} and Kyuheon Kim^{a)†}

요 약

최근 N 스크린 (n-screen) 과 secondary device이란 말이 사람들의 입에 자주 오르내린다. 또한 주위에서 자주 사용하는 기기를 통해 앞에서 말한 서비스를 직접 경험할 수도 있다. 하나의 멀티미디어 기술이면서 서비스 이름인 N 스크린이란 하나의 콘텐츠를 여러 개의 기기에서 끊김 없이 연속적으로 즐길 수 있는 기술 또는 서비스이다. 그리고 secondary device란 주된 콘텐츠를 보여주는 main device와 연동하여 주된 콘텐츠에 관련된 추가 콘텐츠를 보여주는 보조 기기가 있는 서비스이다. 이러한 기술 및 서비스가 각 이동통신사 등에서 몇 년 전부터 거론되어 오고 있으며 현재 대중화된 다양한 멀티미디어 기기에 재조명되어 제공되고 있다. 그러나 현재의 이 서비스는 단순히 사용자가 정식 구매한 멀티미디어 콘텐츠를 자신의 기기가 아닌 이동통신사의 미디어 서버에 올려놓고 필요에 따라 인터넷을 통해 접근한다는 한계가 있다. 이에 본 논문에서는 장면의 동적인 업데이트를 위해 다양한 명령을 포함한 장면 서술정보를 제공 할 수 있는 국제 표준 규격인 MPEG-4 LAsER (Lightweight Application Scene Representation) PMSI (Presentation and Modification of Structured Information) 를 활용하여 사용자가 좀 더 효과적이고 다양하게 N 스크린 혹은 secondary device 서비스를 사용할 수 있는 방안을 제안한다.

Abstract

Recently, N-screen service and secondary device have gotten an attention from public. Also, we can experience N-screen service through various digital devices. N-screen means multimedia technology which can seamlessly consume multimedia content. Secondary device means auxiliary multimedia device which can consume content related to main content through adjunct connection to main device. Not only be electronic manufactures interested in N-screen technology and services but also digital devices applied for N-screen have been released. But it has limitation that user can only consume content to be purchased from content company server not device of user. This paper proposes the system that composes effective and various N-screen multimedia service through MPEG-4 LAsER (Lightweight Application Scene Representation) PMSI (Presentation and Modification of Structured Information) as international standard technology which can provide scene description including many instruction for dynamic update of scene.

Keyword : MPEG-4 LAsER, PMSI, n-screen, secondary device, synchronization

I. 서론

과거에는 TV나 PC 각각의 기기 스크린에 종속적으로 서비스가 제공되었었다면, 최근에는 다양한 스마트 기기들이 출시되면서 동영상과 같은 멀티미디어 콘텐츠를 TV나 PC, 모바일 등의 여러 스크린에서 이용할 수 있는 N 스크린(n-screen) 서비스 환경이 주어지고 있다. N 스크린 서비스의 시작은 10여 년 전으로 거슬러 올라가는데 그 당시에는 N 스크린 서비스라기보다는 평범한 인터넷 동영상 서비스에 가까웠다. 콘텐츠를 공급하는 회사들은 불법복제와 DVD 판매 수익 등 많은 요인으로 거부 반응을 보였다. 그 당시 인터넷 망은 아직 동영상을 제대로 보여주기에 충분하지 않았으며, 컴퓨터를 제외한 기기들은 동영상을 보기에 적당하지 않았다. 하지만 그로부터 10여 년이 지나고 인터넷을 기반으로 거의 모든 콘텐츠가 디지털로 제작되고, 디지털로 유통되는 환경이 되면서 이제 굳이 오프라인 유통방식으로 콘텐츠를 공급할 필요가 점점 없어지면서 또한 사람들의 소비 습관이 변함에 따라 온라인으로 콘텐츠를 공급해야만 하는 상황으로 시장의 변화가 시작 되었다. 거기에 초고속 인터넷망을 비롯하여, 무선 인터넷망도 점점 동영상을 보기에 불편하지 않을 정도로 빨라져 클라우드 서비스를 통해 콘텐츠를 제공해도 될 만해 졌고, 스마트폰 및 태블릿 PC의 등장으로 TV 앞에 앉아있는 사람들의 숫자는 점점 줄어만 가고 있는 것이 현실이 되었다. 이와 같은 이유로 N 스크린 서비스는 주목을 받고 있는 것이다.

위의 N 스크린 서비스가 주목 받는 이유에서도 나왔듯이 N 스크린 서비스를 구현하기 위해서는 전제 조건이 몇 가지 있다. 첫째, 콘텐츠 접근 권한에 대한 보안 문제의 해결이다. 둘째, 사용자의 다양한 기기나 서비스 환경의 차이에

도 불구하고 동일하게 구현되기 위한 표준화이다. 셋째, 무선 모바일 환경에서도 콘텐츠 스트리밍에 제약이 없는 빠른 네트워크이다. 마지막으로 어떤 기기도 접근 가능하며 앞선 기기의 스트리밍 및 서비스 정보를 기록하고 있는 대용량의 저장 공간이다. 현재 이런 전제 조건을 충족하여 N 스크린 서비스가 상용화 되고 있긴 하지만 콘텐츠의 한정적인 접근 문제 및 제한적인 업데이트 문제가 존재한다.

이러한 문제를 해결하기 위해 본 논문에서는 MPEG-4 LAsEr (Lightweight Application Scene Representation) 기술을 활용한다. 국제 표준 규격인 MPEG-4 LAsEr에는 장면의 동적인 업데이트를 위해 다양한 명령을 포함한 장면 서술정보를 제공하고 있으며, 또한 구조화된 정보의 표현 및 수정 방법으로 PMSI (Presentation and Modification of Structured Information) 기술을 포함하고 있다. 본 논문에서는 LAsEr PMSI를 통하여 N 스크린 서비스에서 사용되는 각종 콘텐츠의 효율적인 접근 방법을 보여주고 LAsEr의 여러 가지 이벤트를 포함한 다양한 명령을 활용한 콘텐츠의 효과적인 업데이트 방법을 소개하고자 한다. 본 논문에서의 실험 방법은 파노라믹 영상의 일부를 main device에서 보여주고 나머지 부분을 secondary device에서 보여주는데 이때 main device에서 발생한 슬라이드 쇼 이벤트를 통하여 secondary device에서도 슬라이드 쇼가 진행되게 하는 것이다.

II. MPEG-4 LAsEr

II장에서는 N 스크린 서비스에 접목시킬 MPEG-4 LAsEr 기술에 대해 기술한다. II장 1절에서는 MPEG-4 LAsEr 기술의 개발 배경 및 목적에 대해 기술한다. II장 2절에서는 MPEG-4 LAsEr 기술의 기능과 구조에 대해 기술한다. II장 3절에서는 MPEG-4 LAsEr PMSI 기술에 대해 기술한다.

1. LAsEr 기술 개발 배경 및 목적

정보통신 기술의 발전에 따라 사용자는 이제 단순한 AV 위주의 멀티미디어 서비스에서 벗어나 혼합된 여러 가지의

a) 경희대학교 전자정보학부(College of Electronics and Information, Kyung Hee University)

‡ Corresponding Author : 김규현 (Kyuheon Kim)

E-mail: kyuheonkim@khu.ac.kr

Tel: +82-2-3408-3753 Fax: +82-2-3408-4321

※ 이 논문은 지식경제부의 지원을 받는 지식경제 기술혁신사업("스케일러블 데이터기반 Edutainment 핵심기술 개발")(과제번호 : 10035486)의 연구 결과와 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업(NIPA-2012- H0301-12-1006)의 연구결과로 수행되었음.

Manuscript received June 13, 2012 Revised August 30, 2012

Accepted October 9, 2012

미디어가 사용자의 요구사항을 받아들이는 다양한 대화형 서비스를 제공하는 형식으로 발전하고 있다^[1]. 이러한 기술을 리치미디어 (Richmedia) 서비스라 하며 서비스 기술 개발이 발전하고 확장됨에 따라 통신과 방송 영역에서 활용될 기술 개발이 필요하게 되었다.

이러한 리치미디어 서비스는 기존의 MPEG-4에서 제공하는 장면 구성 기술인 Binary Format for Scene (BIFS)를 통해 가능하였다^[1]. 지상파 DMB에서 TV 프로그램과 데이터 서비스를 결합시키는 BIFS는 여러 개의 객체 (영상화면, 텍스트 화면 등)에 대해 각 객체가 어디에 위치할지 언제 등장할 지를 설정해 두는 방식이다. 즉 TV 화면에 다른 데이터를 얻을 수 있고, 이 데이터를 선택하면 다른 정보 화면이 뜨게 하는 식으로 연동형 데이터서비스를 가능케 하는 기술이다. 그러나 BIFS의 경우, 해당 기술이 Digital Television (DTV)의 Set Top Box (STB) 또는 인터넷 방송을 그 서비스 대상으로 여기고 있었기에 그 복잡도가 상당히 높아 연산능력이 충분한 단말에 적용된 기술로 메모리 용량이나 저장 용량과 성능이 부족한 휴대폰 같은 모바일 기기에서는 한계가 있다.

이를 보완하기 위해 MPEG은 풍부한 화면 구성과 압축의 효율성 두 가지를 모두 고려한 새로운 규격을 만들었고 이것이 바로 LASEr이다^[2]. LASEr는 2004년부터 본격적인 표준화 작업이 진행되었으며, 2006년 1월 첫 번째 버전의 표준 작업이 완료되었다. 또한 2008년과 2009년에는 멀티미디어가 다양한 종류의 단말기에서 소비되는 환경을 고려하여, 하나의 콘텐츠를 다양한 단말에 맞춤형 서비스를 제공하는 Scene Adaptation 기능을 추가 하였고, 2009년에는 장면서술정보에서 구조화된 정보의 일부분을 참조하여 장면을 효과적으로 표현해주는 PMSI 기술을 LASEr에 추가하는 작업을 진행하여 표준 문서 작업은 완료하였다. 또한 현재 2010년에는 위의 PMSI 기술을 접목시킨 LASEr 기능을 점검하는 Reference Software 작업을 하였다.

2. 장면 기술 정보 표준 LASEr

LASEr는 멀티미디어 서비스를 위한 장면 기술 정보를 바이너리 형식으로 제공하는 방법을 정의한 규격으로써,

<그림 1>에서 나타나는 바와 같이 크게 멀티미디어 장면 기술 부분과 미디어로 구성되어 있다. LASEr를 구성하는 미디어로는 정지 영상, 동영상, 오디오 및 폰트를 지원하며, 장면 기술은 텍스트, 그래픽스, 애니메이션과 각각의 객체인 미디어에 대한 시공간적 구성 정보와 대화형 기능 등으로 구성되어 있다^[2].

이러한 장면 구성 정보는 즉 LASEr 장면 기술은 크게 아래와 같이 3가지 기능제공을 목적으로 한다.

- 1) 미디어 및 그래픽스와 같은 장면을 구성하는 요소들의 시공간적 구성 방법
- 2) 하나의 장면을 구성하는 요소 간의 상호 작용 방법
- 3) 장면이 변경될 경우, 변경 부분을 반영하는 방법

상기의 기능을 제공하기 위하여 LASEr에서는 World Wide Web Consortium (W3C)의 주도하에 개발된 오픈 표준의 벡터 그래픽 파일 형식인 Scalable Vector Graphics (SVG)에 기반을 두어 장면을 구성하며, SVG 규격에서 지원하지 않은 기능들은 LASEr extensions를 통해 지원하도록 하였다. 초기에 LASEr는 SVG 1.1 버전을 기반으로 하였지만, SVG profile에 비해 양은 작지만 device GUI, 2D Game, 대기화면 서비스 등과 같은 Visual Application 적용을 위한 리치미디어와 강력한 Interactivity를 지원하는 모바일 버전의 SVG Tiny 1.2버전을 수용하였다^[3].

그림 1. LASEr와 SAF 구조

Fig. 1. structure of LASEr and SAF

또한 SVG는 XML로 기술하므로 웹 브라우저 상에서 열람할 수 있고, 문서 편집기 등에서 편집을 할 수 있는 특징이 있으며, 하이퍼링크를 이미지에 걸거나, 자바스크립트 등과 연동시킬 수 있는 기능 등을 포함하고, 벡터 그래픽스 이므로 확대나 축소를 해도 화질에 변화가 없는 장점을 가지고 있다.

그림 2. SVG 장면 구성 요소
Fig. 2. components of SVG scene

이러한 SVG의 장면 구성요소는 <그림 2>에서 보는 바와 같이 도형, 텍스트, 이미지, 등의 다양한 조합과 애니메이션, 하이퍼링크, interactivity, scripting등의 기능을 XML 기반의 파일 형식으로 표현한 것이다^[4].

따라서 LAsER extensions은 상기의 SVG의 장면 구성 요소에 주로 interactivity와 A/V 콘텐츠의 구성요소를 표현하기 위해 필요한 장면 구성 요소와 좀 더 다양한 기능을 제공하기 위해 추가한 것으로 볼 수 있다.

LAsER는 위의 장면 구성 요소들을 MPEG-4 BIFS에서 활용중인 ‘command’ 방식을 활용함으로써 현재 보이고 있는 장면에 나타난 미디어 객체를 삭제/변경할 수 있으며, 또한 새로운 미디어 객체를 삽입할 수 있게 함으로써 보다 효율적으로 현재 장면에 대한 변경을 가능하게 한다.

또한 LAsER는 ‘command’ 형식으로 구성된 장면 요소를 스트리밍 서비스에 적합하게 이진화 하는 과정을 거쳐, 별도의 포맷 형식인 Simple Aggregation Format (SAF) 이라는 전송 규격 형태로 정의하여 전송을 한다.

SAF 규격은 LAsER 콘텐츠를 장면 묘사와 각 구성 미디

그림 3. SAF Packet architecture
Fig. 3. SAF Packet architecture

어 요소를 포함한 포괄적 데이터의 이진화된 비트스트림을 전송하기 위한 전송 규격으로써 모바일 환경과 같은 작은 전송 대역폭에 적합하게 전송하기 위하여 다중화에 필요한 헤더 정보를 <그림 3>에서 보는 바와 같이 최적화하여 제공한다^[2].

마지막으로 LAsER 콘텐츠는 MPEG-4의 ISO Base Media File Format을 기반으로 하는 MP4 file 형태로 저장된다. 이 파일 포맷은 다른 여러 파일 포맷기술의 기본 형태이며 가장 일반적인 포맷 기술로 각각의 객체를 track별로 기술하여 저장하기에, 위의 장면기술은 video, audio, text 등과 같이 각각의 media와는 별개로 scene track에 저장된다. 이 장면기술을 포함하는 track은 미디어의 타입을 나타내는 handler_type이 “sdsm”이라는 장면기술타입 (Scene description type)의 값을 가지고, LAsER Header정보는 각 트랙의 헤더 정보를 기술하는 ConfigurationBox에 기술함으로써 이진화된 LAsER 스트림을 파일 포맷 형태로 저장할 수 있다.

3. LAsER PMSI 기술

PMSI는 장면표현기술을 활용하여 구조화된 메타데이터를 표현하는 기술로서, 구조화된 정보 (메타데이터)를 일정한 방식으로 장면상에 시각적으로 표현할 수 있게 해준다. 아래의 <표 1>은 구조화된 메타데이터를 표현하기 위한 PMSI의 주요 구성요소이다^[5].

표 1. MPEG-4 LAsER의 PMSI 구성요소

Table 1. components of MPEG-4 LAsER PMSI

Components	Functionality
mpeg-pmsi() scheme	XPath 기반 접근을 이용한 Pointing scheme
SVG tref 요소	텍스트 콘텐츠 참조
LAsER externalReference 요소	주기적으로 요소와 속성들을 업데이트
LAsER externalUpdate 요소	미리 정의된 구조화 정보 수정

1.1 PI (Presentation Information) 와 SI (Structured Information)

PI는 "어떻게" 각 객체를 장면에 보여줄지 선언하기 위한 정보로서 다양한 객체들의 시공간적인 배치가 명시되어 있는 장면 서술 정보이다. 그리고 SI는 "무엇을" 장면에 보여줄지 선언되어 있는 서술 정보이다.

1.2 PMSI 구성 요소

1.2.1 mpeg-pmsi() scheme

mpeg-pmsi() scheme는 포인터 메커니즘으로서, 스키마-기반 XML 문서에 서술된 어떠한 resource에도 접근할 수 있는 속성을 지닌 W3C의 XPath를 지원한다. 또한 XPath처럼 주어진 노드 집합에서 원하지 않는 노드를 필터링 시킬 수도 있다. 이것은 <표 1>에서도 pointing scheme라고 기술되어 있듯이 말 그대로 무엇인가를 가리키는 정보를 가지고 있다. PI는 SI를 SI는 사용할 콘텐츠를 각각 가리키고 있다.

1.2.2 SVG tref 요소

PMSI application에서 XML문서에 서술된 textual content는 가장 빈번하게 참조되는 정보이다. textual content는 실제 문자 데이터 또는 참조된 요소의 문자 데이터이다. 이러한 텍스트 정보들을 가져오기 위해서는 SVG tref 요소와

표 2. SVG tref 요소의 사용 예

Table 2. example of SVG tref element

```

<?xml version='1.0' encoding='EUC-KR'?>
<saf:SAFSession xmlns:saf='urn:mpeg:mpeg4:SAF:2005' xmlns:lsr='urn:mpeg:mpeg4:LAsER:2005'
xmlns='http://www.w3.org/2000/svg' xmlns:xlink='http://www.w3.org/1999/xlink' xmlns:smil='http://www.w3.org/2005/SMIL21'
xmlns:ev='http://www.w3.org/2001/xml-events'>
  <saf:sceneHeader>
    <lsr:LAsERHeader profile='full' resolution='7' timeResolution='1000' coordBits='19'
colorComponentBits='8' useFullRequestHost='true' hasStringIDs='true'/>
  </saf:sceneHeader>
  <saf:sceneUnit>
    <lsr:NewScene>
      <svg id='root' width='320' height='240' viewport-fill='white'>
        <g>
          <text id='text' x='87' y='89' font-family='Gulim' font-size='10'
font-weight='bold' fill='#000000' fill-opacity='1.00'><tref xlink:href='http://in-
tmedia.khu.ac.kr/pmsi/tref/test.xml#pmsi(/mpeg7:FreeTextAnnotation/text()'/></text>
        </g>
      </svg>
    </lsr:NewScene>
  </saf:sceneUnit>
</saf:SAFSession>

```

함께 xlink 속성을 내포하는 것이 요구된다. SVG 에 있는 text 요소와 textArea 요소를 이용한다면 XML 문서의 텍스트 내용을 보여주는 것을 기대할 수 있다. 따라서 SVG 1.1 tref 요소는 MPEG-4 LAsER 표준에 포함되었으며, SVG text와 SVG textArea의 자식요소로 쓰일 수 있도록 제한되었다. <표 2>는 SVG tref 요소가 XML 문서에서 어떻게 표현되고 이용되는지 보여주고 있다. tref의 xlink:href 속성 값으로 SI를 가리키고 있다. 따라서 tref로 가리키고 있는 문자 데이터를 바로 위의 text 요소에 기술 되어 있는 PI로 scene에 보여주게 된다.

1.3 LAsER externalReference 요소

SI는 주기적으로 자기 자신을 변화 시킬 수 있어야 한다. 따라서 SI의 업데이트를 주기적으로 반영하여 표현하는 메커니즘이 반드시 필요하다. 해당 메커니즘을 위해 PMSI에

서는 주기적으로 업데이트 되는 장면에 정해진 시간간격에서 최신 SI 버전을 제휴시키는 새로운 컨테이너 요소인 LAsER externalReference를 정의하였다. <표 3>은 LAsER externalReference 요소가 XML 문서에서 어떻게 표현되고 이용되는지 보여주고 있다. externalReference의 updateInterval 속성 값을 주기로 하여 SI의 업데이트를 주기적으로 반영한다.

1.4 LAsER externalUpdate 요소

PMSI application에서는 presentation의 결과를 SI에 반영하는 메커니즘이 요구된다. 예를 들어 현재의 화면에서 이용자가 자신의 선호 항목(장르, 색깔, 위치)을 선택하면 이용자 선호 데이터를 포함한 SI는 이용자 특성 정보를 반영하여 업데이트 되어야 한다. 따라서 presentation 쪽에서 SI를 변경할 수 있는 메커니즘이 필요하다. LAsER ex-

표 3. LAsER externalReference 요소의 사용 예
Table 3. example of LAsER externalReference element

```
<?xml version='1.0' encoding='EUC-KR'?>
<saf:SAFSession xmlns:saf='urn:mpeg:mpeg4:SAF:2005' xmlns:lsrc='urn:mpeg:mpeg4:LAsER:2005'
xmlns='http://www.w3.org/2000/svg' xmlns:xlink='http://www.w3.org/1999/xlink' xmlns:smil='http://www.w3.org/2005/SMIL21'
xmlns:ev='http://www.w3.org/2001/xml-events'>
  <saf:sceneHeader>
    <lsrc:LAsERHeader profile='full' resolution='7' timeResolution='1000' coordBits='19' colorComponentBits='8'
useFullRequestHost='true' hasStringIDs='true'/>
  </saf:sceneHeader>
  <saf:sceneUnit>
    <lsrc:NewScene>
      <svg id='root' width='320' height='240' viewport-fill='white'>
        <g>
          <lsrc:externalReference id='updateText' updateInterval='30'>
            <text id='s4' x='87' y='89' font-family='Gulim' font-size='10' font-weight='bold' fill='#000000' fill-opac-
ity='1.00'><tref xlink:href='http://intmedia.khu.ac.kr/pmsi/externalRef/
test_DI3.xml#pmsi(/mpeg7:FreeTextAnnotation/text())'/></text>
          </lsrc:externalReference>
        </g>
      </svg>
    </lsrc:NewScene>
  </saf:sceneUnit>
</saf:endOFSAFSession>
</saf:SAFSession>
```

표 4. LAsEr externalUpdate 요소의 사용 예

Table 4. example of LAsEr externalUpdate element

```

<?xml version='1.0' encoding='EUC-KR'?>
<saf:SAFSession xmlns:saf='urn:mpeg:mpeg4:SAF:2005' xmlns:lsr='urn:mpeg:mpeg4:LAsEr:2005'
xmlns='http://www.w3.org/2000/svg' xmlns:xlink='http://www.w3.org/1999/xlink' xmlns:smil='http://www.w3.org/2005/SMIL21'
xmlns:ev='http://www.w3.org/2001/xml-events'>
  <saf:sceneHeader>
    <lsr:LAsErHeader profile='full' resolution='7' timeResolution='1000' coordBits='19' colorComponentBits='8'
useFullRequestHost='true' hasStringIDs='true'/>
  </saf:sceneHeader>
  <saf:sceneUnit>
    <lsr:NewScene>
      <svg id='root' width='320' height='240' viewport-fill='white'>
        <g>
          <lsr:externalUpdate id='go1' updateInterval='none' type='insert' xlink:href='http://in-
tmedia.khu.ac.kr/pmsi/EU/Add/test_DI_add.xml#pmsi(/mpeg7:Creation/mpeg7:Title[@type='main'])' attributeName='textContent'
value='Step3-'/>
          <ev:listener event='accessKey(2)' handler='#go1'/>
        </g>
      </svg>
    </lsr:NewScene>
  </saf:sceneUnit>
</saf:SAFSession>

```

ternalUpdate element는 SI의 업데이트를 지원하며, 업데이트는 insert, replace, delete 세 가지 경우를 지원한다. <표 4>는 LAsEr externalUpdate 요소가 XML 문서에서 어떻게 표현되고 이용되는지 보여주고 있다. externalUpdate의 type 속성 값에 따라 value 속성 값을 이용하여 업데이트 된다.

III. 신규방안

III장에서는 MPEG-4 LAsEr PMSI를 이용하여 N 스크린 서비스에서 사용되는 콘텐츠의 효율적인 접근 방법과 다양한 명령을 활용한 콘텐츠의 효과적인 업데이트 방법을 제안한다. 이때 방안 검토에 사용된 콘텐츠는 이벤트에 의한 슬라이드 쇼로 하였다.

1. 이벤트에 의한 슬라이드 쇼 콘텐츠

기존의 N 스크린 서비스에서 사용되는 콘텐츠는 일반적으로 동영상이다. 사용자가 기기를 변경하여 다른 스크린을 통해서 끊임 없이 연속적으로 동영상을 계속 볼 수 있는 시나리오이다. 그러기 위해서는 우선 비디오 스트리밍 전송과 비디오 프레임별 조작이 가능한 플레이어 혹은 렌더러가 있어야한다. 하지만 본 논문의 목적은 비디오의 프레임별 스트리밍 서비스가 가능한지를 보여주는 것이 아니라 기존의 N 스크린 서비스에 콘텐츠의 효율적인 접근 방법과 다양한 명령을 활용한 콘텐츠의 효과적인 업데이트 방법이 가능하다는 것을 보여주기 위함이다. 따라서 이벤트에 의한 슬라이드 쇼 콘텐츠를 이용하였다.

위의 <그림 4>는 MPEG-4 LAsEr PMSI를 이용하여 N 스크린 서비스를 구현하는데 사용될 원본 파노라믹 이미지

그림 4. 원본 파노라믹 이미지
Fig. 4. original panoramic image

이다. 이 파노라믹 이미지는 3000*960px의 크기로 일반 기기에서는 한 번에 스크린에 보여주기가 힘들다. 따라서 main device에 이미지의 일부를 보여주고 secondary device을 연결시켰을 때 남은 이미지가 보이게 된다.

2. 구현 환경

Windows XP 혹은 7의 OS에 Microsoft Visual studio 2008를 사용하여 소프트웨어를 개발하였고 실행하였다.

위의 <그림 5>는 아래에서 설명할 시스템의 구현 환경이다. 이 시스템은 server, main device, secondary device로 구성되어 있다. server에 있는 SI는 사용자가 사용하게 될 콘텐츠의 위치 정보를 가지고 있다. 그리고 main device와 secondary device에는 사용할 SI의 위치 정보인 PI가 있다. PI를 통해 주기적으로 SI를 가지고 오다가 main device에서 발생한 event에 의해 변화된 SI에 따라 새로운 콘텐츠를 사용자가 사용하게 된다. event는 main device에서만 발생하

그림 5. 구현 환경
Fig. 5. setting for execution

지만 그 event에 의해 server의 SI가 바뀌게 되고 secondary device에서는 주기적으로 SI를 가져옴으로써 바뀐 SI가 자연스럽게 반영된다.

2.1 서버와 SI

N 스크린 서비스의 클라우드 저장공간에 해당하는 서버에 원본 이미지와 사용자가 "무엇을" 장면에 보여줄지 선언되어 있는 서술 정보인 SI를 위치시킨다.

위의 <표 3>은 xml로 기술된 SI로 사용자에게 보일 이미지의 위치 정보를 가지고 있다. 위에 표시된 부분은 각각 main device와 secondary device에서 사용할 이미지를 각각

가리키고 있는 것이다.

2.2 main device와 PI

사용자가 다루는 main device에는 앞에서 말한 서버와 SI의 위치 정보를 가지고 있는 PI를 위치시킨다.

위 <표 4>는 xml로 기술된 PI로 SI의 위치 정보를 가지고 있다. 또한 main device에 “어떻게” 보여줄지에 대한 시간적인 정보와 슬라이드 쇼를 컨트롤할 이벤트 정보가 기술되어 있다. 이때 PI는 사용자가 일부 조작이 가능하다. 예를 들어 이벤트의 종류를 달리하여 자신이 원하는 형태로 슬라이드 쇼를 진행시킬 수 있다.

표 5. xml로 기술된 SI
Table 5. SI by xml

<?xml version="1.0" encoding="UTF-8"?>	
- 종략 -	
<DescriptionMetadata>	
<LastUpdate>2003-06-19T00:22:45</LastUpdate>	
<CreationTime>2003-06-19T00:22:45</CreationTime>	
<Instrument>	
<Tool>	
<Name>XMLSPY 5.0</Name>	
</Tool>	
</Instrument>	
</DescriptionMetadata>	
<Description xsi:type="CreationDescriptionType">	
<CreationInformation>	
<Creation>	
<Title type="main" xml:lang="en">main image</Title>	
<TitleMedia>	
<MainImage>	
<MediaUri>../Resource/a1.png</MediaUri>	
</MainImage>	
</TitleMedia>	
- 종략 -	
<Title type="second" xml:lang="en">secondary image</Title>	
<TitleMedia>	
<SecondaryImage>	
<MediaUri>../Resource/b1.png</MediaUri>	
</SecondaryImage>	
</TitleMedia>	
- 이하 생략 -	

표 6. xml로 기술된 PI(main device)

Table 6. PI by xml(main device)

```
<?xml version='1.0' encoding='UTF-8'?>
<saf:SAFSession xmlns:saf='urn:mpeg:mpeg4:SAF:2005' xmlns:lss='urn:mpeg:mpeg4:LAsER:2005'
xmlns='http://www.w3.org/2000/svg' xmlns:xlink='http://www.w3.org/1999/xlink' xmlns:smil='http://www.w3.org/2005/SMIL21'
xmlns:ev='http://www.w3.org/2001/xml-events'>
  <saf:sceneHeader objectTypeIndication='9' streamType='3'>
    <lss:LAsERHeader pathcomponents='15' useFullRequestHost='true' timeResolution='1000' colorComponentBits='8' resolution='7'
coordBits='19'/>
  </saf:sceneHeader>

  <saf:RemoteStreamHeader streamID='Stream_a' objectTypeIndication='108' streamType='4' source='http://in-
tmedia.khu.ac.kr/Resource/si.xml#pmsi(//TitleMedia/MainImage/MediaUri/text())'/>

  <saf:sceneUnit>
    <lss:NewScene>
      <svg id='svg' height='960' viewBox='0 0 1500 960' width='1500'>

        <g>
          <lss:externalReference id='eR' updateInterval='1'>
            <image id='a' height='960' type='image/jpg' width='1500' x='0' y='0' xlink:href='stream:Stream_a'/>
          </lss:externalReference>

          <lss:externalUpdate id='eU_a1' updateInterval='none' type='replace' attributeName='textContent' value='http://in-
tmedia.khu.ac.kr/Resource/a1.png' xlink:href='http://in-
tmedia.khu.ac.kr/Resource/si.xml#pmsi(//TitleMedia/MainImage/MediaUri/text())'/>
          <lss:externalUpdate id='eU_a2' updateInterval='none' type='replace' attributeName='textContent' value='http://in-
tmedia.khu.ac.kr/Resource/a2.png' xlink:href='http://in-
tmedia.khu.ac.kr/Resource/si.xml#pmsi(//TitleMedia/MainImage/MediaUri/text())'/>
          <lss:externalUpdate id='eU_a3' updateInterval='none' type='replace' attributeName='textContent' value='http://in-
tmedia.khu.ac.kr/Resource/a3.png' xlink:href='http://in-
tmedia.khu.ac.kr/Resource/si.xml#pmsi(//TitleMedia/MainImage/MediaUri/text())'/>
          <lss:externalUpdate id='eU_a4' updateInterval='none' type='replace' attributeName='textContent' value='http://in-
tmedia.khu.ac.kr/Resource/a4.png' xlink:href='http://in-
tmedia.khu.ac.kr/Resource/si.xml#pmsi(//TitleMedia/MainImage/MediaUri/text())'/>
          <lss:externalUpdate id='eU_a5' updateInterval='none' type='replace' attributeName='textContent' value='http://in-
tmedia.khu.ac.kr/Resource/a5.png' xlink:href='http://in-
tmedia.khu.ac.kr/Resource/si.xml#pmsi(//TitleMedia/MainImage/MediaUri/text())'/>
          <lss:externalUpdate id='eU_a6' updateInterval='none' type='replace' attributeName='textContent' value='http://in-
tmedia.khu.ac.kr/Resource/a6.png' xlink:href='http://in-
tmedia.khu.ac.kr/Resource/si.xml#pmsi(//TitleMedia/MainImage/MediaUri/text())'/>

          <lss:externalUpdate id='eU_b1' updateInterval='none' type='replace' attributeName='textContent' value='http://in-
tmedia.khu.ac.kr/Resource/b1.png' xlink:href='http://in-
tmedia.khu.ac.kr/Resource/si.xml#pmsi(//TitleMedia/SecondaryImage/MediaUri/text())'/>
        </g>
      </svg>
    </lss:NewScene>
  </saf:sceneUnit>
</saf:RemoteStreamHeader>
</SAFSession>
```

```

<lsr:externalUpdate id='eU_b2' updateInterval='none' type='replace' attributeName='textContent' value='http://in-
tmedia.khu.ac.kr/Resource/b2.png' xlink:href='http://in-
tmedia.khu.ac.kr/Resource/si.xml#pmsi(//TitleMedia/SecondaryImage/MediaUri/text())' />
<lsr:externalUpdate id='eU_b3' updateInterval='none' type='replace' attributeName='textContent' value='http://in-
tmedia.khu.ac.kr/Resource/b3.png' xlink:href='http://in-
tmedia.khu.ac.kr/Resource/si.xml#pmsi(//TitleMedia/SecondaryImage/MediaUri/text())' />
<lsr:externalUpdate id='eU_b4' updateInterval='none' type='replace' attributeName='textContent' value='http://in-
tmedia.khu.ac.kr/Resource/b4.png' xlink:href='http://in-
tmedia.khu.ac.kr/Resource/si.xml#pmsi(//TitleMedia/SecondaryImage/MediaUri/text())' />
<lsr:externalUpdate id='eU_b5' updateInterval='none' type='replace' attributeName='textContent' value='http://in-
tmedia.khu.ac.kr/Resource/b5.png' xlink:href='http://in-
tmedia.khu.ac.kr/Resource/si.xml#pmsi(//TitleMedia/SecondaryImage/MediaUri/text())' />
<lsr:externalUpdate id='eU_b6' updateInterval='none' type='replace' attributeName='textContent' value='http://in-
tmedia.khu.ac.kr/Resource/b6.png' xlink:href='http://in-
tmedia.khu.ac.kr/Resource/si.xml#pmsi(//TitleMedia/SecondaryImage/MediaUri/text())' />

```

(event에 의해 어떻게 SI를 변경시킬지에 대한 정보)

```

<ev:listener event='accessKey(1)' handler='#eU_a1' enabled='true' />
<ev:listener event='accessKey(2)' handler='#eU_a2' enabled='true' />
<ev:listener event='accessKey(3)' handler='#eU_a3' enabled='true' />
<ev:listener event='accessKey(4)' handler='#eU_a4' enabled='true' />
<ev:listener event='accessKey(5)' handler='#eU_a5' enabled='true' />
<ev:listener event='accessKey(6)' handler='#eU_a6' enabled='true' />

<ev:listener event='accessKey(1)' handler='#eU_b1' enabled='true' />
<ev:listener event='accessKey(2)' handler='#eU_b2' enabled='true' />
<ev:listener event='accessKey(3)' handler='#eU_b3' enabled='true' />
<ev:listener event='accessKey(4)' handler='#eU_b4' enabled='true' />
<ev:listener event='accessKey(5)' handler='#eU_b5' enabled='true' />
<ev:listener event='accessKey(6)' handler='#eU_b6' enabled='true' />

```

(설정된 event)

```

</g>

</svg>
</lsr:NewScene>
</saf:sceneUnit>
</saf:endOfSAFSession>
</saf:SAFSession>

```

2.3 secondary device와 PI

main device에 연결시키는 secondary device에도 서버와 SI의 위치 정보를 가지고 있는 PI를 위치시킨다.

위 <표 5>는 xml로 기술된 PI로 SI의 위치 정보를 가지고 있다. 또한 secondary device에 “어떻게” 보여줄지에

대한 시공간적인 정보가 기술되어 있다. 다만 main device의 PI와 다른 점은 슬라이드 쇼를 컨트롤할 이벤트 정보가 없다는 점이다. main device를 통해 슬라이드 쇼가 컨트롤되고 <표 5>의 ‘lsr:externalReference’를 이용하여 주기적으로 동기화되어 secondary device에 업데이트

표 7. xml로 기술된 PI(secondary device)

Table 7. PI by xml(secondary device)

<pre> <?xml version='1.0' encoding='UTF-8'?> <saf:SAFSession xmlns:saf='urn:mpeg:mpeg4:SAF:2005' xmlns:lsr='urn:mpeg:mpeg4:LAsER:2005' xmlns='http://www.w3.org/2000/svg' xmlns:xlink='http://www.w3.org/1999/xlink' xmlns:smil='http://www.w3.org/2005/SMIL21' xmlns:ev='http://www.w3.org/2001/xml-events'>   <saf:sceneHeader objectTypeIndication='9' streamType='3'>   <lsr:LAsERHeader pathcomponents='15' useFullRequestHost='true' timeResolution='1000' colorComponentBits='8' resolution='7' coordBits='19'/>   </saf:sceneHeader>   <saf:RemoteStreamHeader streamID='Stream_b' objectTypeIndication='108' streamType='4' source='http://in- media.khu.ac.kr/Resource/si.xml#pmsi(//TitleMedia/SecondaryImage/MediaUri/text())'/>   <saf:sceneUnit>   <lsr:NewScene>   <svg id='svg' height='960' viewBox='0 0 1500 960' width='1500'>   <g>   <lsr:externalReference id='eR' updateInterval='1'>   <image id='b' height='960' type='image/jpg' width='1500' x='0' y='0' xlink:href='stream:Stream_b'/>   </lsr:externalReference>   </g>   </svg>   </lsr:NewScene>   </saf:sceneUnit> </saf:endOfSAFSession/> </saf:SAFSession> </pre>

되는 것이다.

3. 실험 결과

본 논문의 실험에서는 파노라믹 장면의 업데이트를 MPEG-4 LAsER PMSI 기술을 활용하여 진행한다.

아래의 <그림 6>은 프로그램의 실행화면이고 <그림 7>은 프로그램의 실행 후 PI 정보가 있는 xml 파일을 선택하는 화면이다. main device와 secondary device 모두 <그림 6>과 <그림 7>의 과정을 거쳐 실행을 하고 <그림 8>과 같이 이미지를 보여지게 된다.

이 때 파노라믹 장면을 하나의 기기에서 다 보여줄 수 없기 때문에 main device에서 파노라믹 장면의 왼쪽 이미

그림 6. 프로그램의 실행

Fig 6. execution of program

그림 7. PI 정보가 있는 xml 파일 선택
Fig. 7. selection of xml file contained PI information

그림 8. 실행 화면
Fig. 8. screen of result

지를 보여주고 나머지 오른쪽 이미지를 secondary device에서 보여주게 된다. 사용자가 직접적으로 조작하는 main device를 통하여 사용자가 event를 발생시키면 secondary device에서는 자동으로 event에 따라 업데이트가 된다.

위의 <그림 9>에서 왼쪽의 이미지는 main device의 스크린이고 오른쪽의 이미지는 secondary device의 스크린이다. 사용자는 main device를 이용하여 이미지를 보다가 남은 이미지를 보기를 위하여 secondary device를 연결시키게 된다. 이 때 secondary device에서는 서버를 통하여 자동으로

그림 9. 이벤트에 의한 슬라이드 쇼
Fig. 9. slideshow by event

업데이트 된 이미지를 보여주게 된다. 사용자는 main device의 이벤트를 통한 컨트롤로 secondary device 까지 컨트롤 할 수 있다.

아래의 <그림 10>은 event에 따른 장면 업데이트가 진행되는 과정을 간략하게 순서도로 나타낸 것이다.

main device의 실행으로 위의 순서도는 시작된다. secondary device의 실행 유무와 상관없이 main device에서는 event가 발생할 수 있으며 그 event에 따라 SI가 변경되어 두 device의 장면은 각각 업데이트가 된다. event가 발생시 secondary device가 실행되지 않은 상황일 때는 차후에 업데이트 된 장면으로 secondary device가 실행이 될 때 보이게 된다.

위의 실험에서는 secondary device에서 파노라믹 이미지의 남은 부분이 보이고 있다. 그리고 main device와 같은 이미지를 보이게 하는 것도 가능하다. 또한 main device에서 보이고 있는 이미지의 설명이 기술된 텍스트 정보나 다른 관련된 이미지나 텍스트를 보이게도 할 수 있다.

그림 10. event에 따른 장면 업데이트가 진행되는 과정
Fig. 10. scene update process by event

IV. 결 론

다양한 멀티미디어 기기들의 출현과 이 기기들을 일상과 업무에 활용하는 사람들이 많아지고 의존도가 높아지면서 PC와 각종 멀티미디어 기기들의 연동이 필요하다. 이 연동을 가능하게 해주는 것이 N 스크린 서비스라고 할 수 있다. 각기 다른 다양한 사양을 가진 기기들의 환경에서 서비스가 가능한 N 스크린은 요즘 큰 관심을 끌고 있고 이동통신업체를 비롯한 여러 콘텐츠 관련 기업에서 N 스크린 사업을 주요 전략으로 삼고 있다. 현재 사용자 입장에서는 N 스크린 서비스 상에서 경험해 볼만한 콘텐츠가 없다는 것이 문제이긴 하지만, 이 문제는 시간이 지나면 해결될 수 있는 문제이긴 하다. N 스크린을 사용하는 사용자가 늘어나고 수익에 대한 확신이 커지면 콘텐츠 관련 기업에서 좀 더 적극적으로 콘텐츠 사업에 발을 들여 놓을 것이기 때문이다.

이런 상황에서 본 논문은 상기에 기술한 N 스크린 서비스 상에서 경험해 볼만한 콘텐츠가 없다는 문제점을 다른 방법으로 해결하기 위한 방안을 제시하는 것이다. N 스크린 서비스에서 MPEG-4 LSeR 기술을 활용하여 하나의 콘텐츠를 다양하고 효과적인 업데이트가 가능하게 하여 사용자의 입

장에서 지루하거나 반복적이지 않은 서비스를 제공할 수 있다. 또한 MPEG-4 LSeR 기술을 활용하면 다양한 이벤트 및 명령이 가능하게 하여 서비스 제공자에게도 도움이 될 것이다. 그리고 MPEG-4 LSeR PMSI 기술을 사용하면 사용자가 직접 콘텐츠에 접근이 가능한 서비스를 제공할 수 있다. 다만 이것을 위해서는 서비스 제공자가 사용자의 부분적인 서버 접근을 가능하게 해줘야 한다는 전제 조건이 있다.

참 고 문 헌

- [1] Kyuheon Kim, Byungchul Kim, *Jihun Cha "Compatible method between LSeR and DIMS for interactive data service" The Institute of Electronics Engineers of Korea 2009. 07
- [2] ISO, IEC 14496-20, Information technology, Coding of audio-visual objects. Part 20: Lightweight Application Scene Representation (LSeR) and Simple Aggregation Format (SAF)
- [3] W3C SVG12, Scalable Vector Graphics (SVG) Tiny 1.2 Specification [Recommendation], <http://www.w3.org/TR/SVGMobile12>
- [4] W3C SVG11, Scalable Vector Graphics (SVG) 1.1 Specification [Recommendation], <http://www.w3.org/TR/2003/REC-SVG11-20030114>
- [5] Injae Lee, Seungwon Song, Hankyu Lee, Jihun Cha, "Rich Media Framework based on MPEG-4 LSeR PMSI Technique" Journal of Broadcast Engineering, vol. 15, no. 2, pp. 248~264, 2010.

저 자 소 개

박 용 철

- 2009년 8월 : 경희대학교 전자공학과 학사
- 2010년 8월 : 경희대학교 전자공학과 석사
- 주관심분야 : 멀티미디어, MPEG-2/4 systems

김 규 현

- 1989년 2월 : 한양대학교 전자공학과 공학사
- 1992년 9월 : 영국 University of Newcastle upon Tyne 전기전자공학과 공학석사
- 1996년 7월 : 영국 University of Newcastle upon Tyne 전기전자공학과 공학박사
- 1996년 ~ 1997년 : 영국 University of Sheffield, Research Fellow
- 1997년 ~ 2006년 : 한국전자통신연구원 대화형미디어연구팀장
- 2006년 ~ 현재 : 경희대학교 전자정보대학 교수
- 주관심분야 : 영상처리, 멀티미디어통신, 디지털 대화형 방송